Programming Project 1 for Introduction to Computer Security

1 Overview

The objective of this assignment is to familiarize you with Bloom Filters and their use for detecting weak password selection. After finishing the assignment, you should be able to select the appropriate design parameters for Bloom Filters needed for weak password detection given a set of bad passwords.

Bloom filters have a lot of useful applications other than weak password detection. Below are some resources on Bloom Filters.

What are bloom filters – Long-winded (and may be funny) explanation with a use case
Bloom Filters by Example - Explanation with a live demo
More examples

2 Submission Guidelines

This project has a total of 100 points.

You need to submit a report answering the questions associated with the task in Section 3. You also need to provide explanations to the observations that are interesting or surprising.

This project requires you to write a program implementing a Bloom Filter. It is recommended you write the program in Python but C is also acceptable. **If you would like to use a different programming language than the two listed above, please contact the TA and get their permission to do so.** Be sure to include a README file on exactly how to compile and/or run your code. Ideally, submissions must be able to run on OSU FLIP servers. **If for whatever reason, this is not possible for your code, please check-in with the TA before starting on your assignment.**

Your submission should contain the code files, a make/README file describing how to compile/run your code, and a PDF document with observations. Submit these on Canvas as a zip/tar folder.

3 Programming Task

[60 pts] You will write a program to identify whether a given set (text input file, see below) of passwords is in a list of known or weak passwords. You will do that by creating a Bloom Filter(s) to check if the given passwords are part of an available dictionary of most commonly used passwords. Some details:

1) *dictionanry.txt* contains a dictionary of common passwords. This list isn't necessarily complete, but it is going to be your set of most common passwords for this assignment to check against. There is one password on each line.
2) *sample_input.txt* is a sample input to your program. The first line is the total number of passwords that your program will check for and then passwords follow from line 2. One password on each line.
3) *sample_output.txt* is what your output should look like. The first line is the total number of passwords that your program checked. Each line is the password followed (space is the delimiter) by either 'no' or 'maybe' (all small characters) depending on whether the password is not in the set or it can be. (This is **NOT** a solution for *sample_input.txt*)
4) You code should accept four inputs as shown: *yourname_bloom_filter -d dictionary.txt -i input.txt -o3 output3.txt -o5 output5.txt*, where *yourname_bloom_filter* is the name of your program (**example shown for C code**). Please create the output3.txt and output5.txt in the current directory if they don't exist.
5) You first use the *dictionary.txt* to create two bloom filters, one using ***3* hash** functions and other using ***5* hash** functions. Then, you run the passwords in *input.txt* through those two bloom filters and return the output in *output3.txt* and *output5.txt* respectively.
6) You are free to choose the appropriate hash functions for your program. The size of the bloom filter will depend on the output range of your chosen hash function.

4 Write-Up Task

[40 pts] Based on above programming task, explain briefly:

1) What hash functions did you choose and why (Hint: Cryptographic or non-cryptographic)?
2) What is the output range of the hash functions and how does it relate to the size of the bloom filter?
3) What is the size of the Bloom filter in each case? Why did you pick this size?
4) How long does it take for your Bloom Filter to check 1 password in each case (3 hashes vs 5 hashes)? Does one Bloom Filter (3 hashed vs 5 hashes) perform better than other? Why or why not?
5) What is the probability of False Positives in your Bloom Filter for each case?

6) How can you reduce the rate of False Positives?
7) What will happen to the False Positive Rate of your 5-hash bloom filter if you set the size of the filter to be the same as that used for 3-hash filter?
8) What is the probability of False Negative in your Bloom Filters?